Experiment Report 3: Hardware Comparison on Tiny Shakespeare (Character-Level nanoGPT)

Experimental Setup

Date: Monday, 02/09/2026
Participants: John Lee, Maximo Sanchez, Youssif Goda

In this experiment, we trained a **tiny character-level GPT model** on the *Tiny Shakespeare* dataset to evaluate whether differences in hardware specifications (CPU vs GPU, and across GPU tiers) lead to observable differences in training behavior at **Large** model scales. Bumping up the heads, layers, and embedding values for extra complexity

All participants trained the same model using the **nanoGPT framework** within the shared https://github.com/MaxiSanc37/GPU-vs-CPU-Comparative-Research repository. To ensure experimental fairness and reproducibility, all runs were logged to a shared Weights & Biases project: **GPUvsCPUResearch** (https://wandb.ai/dickinson-comp560-sp26/GPUvsCPUResearch)

The experiment was conducted on three machines with differing hardware capabilities:

- **RTX 3070 GPU, AMD Ryzen 7 5800H system**
- **RTX 4060 GPU, AMD Ryzen 7 8845HS system**
- **Apple MAC M2 Silicon chip**

These systems were ordered conceptually from highest to lowest theoretical compute capability. All models were trained for the same number of iterations using identical configurations, seeds, and dataset splits.

Model Configuration (Held Constant Across All Runs)

To isolate the effect of hardware, **all hyperparameters were fixed** across devices:

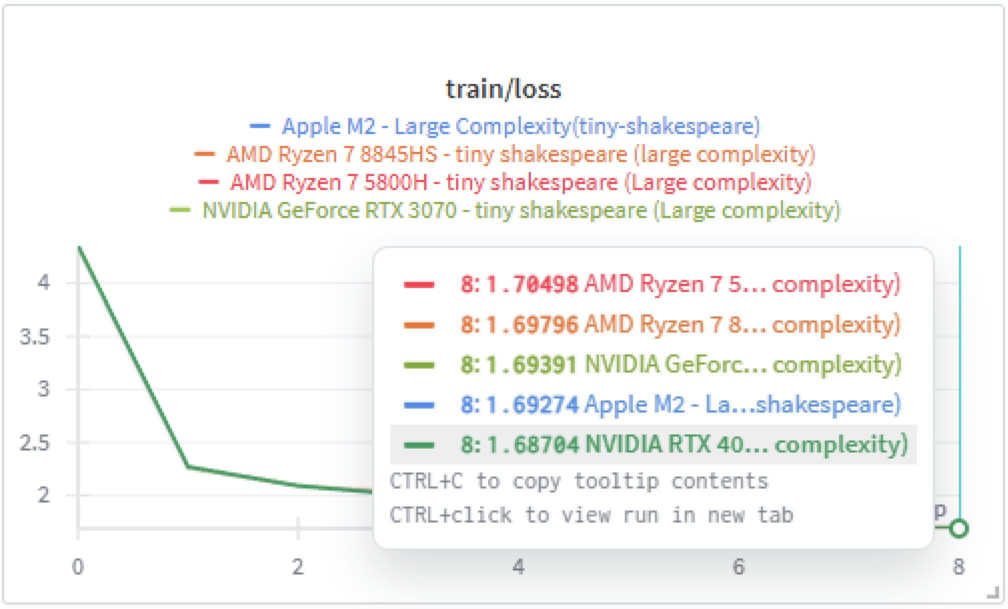
```
# embryonic GPT model
n_layer = 12
n_head = 12
n_embd = 768 # need n_embd % n_head == 0
dropout = 0.0
learning_rate = 1e-3 # with baby networks can afford to go a bit higher
max_iters = 2000
lr_decay_iters = 2000 # make equal to max_iters usually
min_lr = 1e-4 # learning_rate / 10 usually
beta2 = 0.99 # make a bit bigger because number of tokens per iter is small
warmup_iters = 100 # not super necessary potentially
```

We have scaled up to a **Large model (12 layers, 12 heads, 768 embd)**. This configuration represents a '**Transition Zone**'—a mid-sized bridge designed to identify the exact crossover point where GPU efficiency begins to overtake the CPU.

Results and Observations

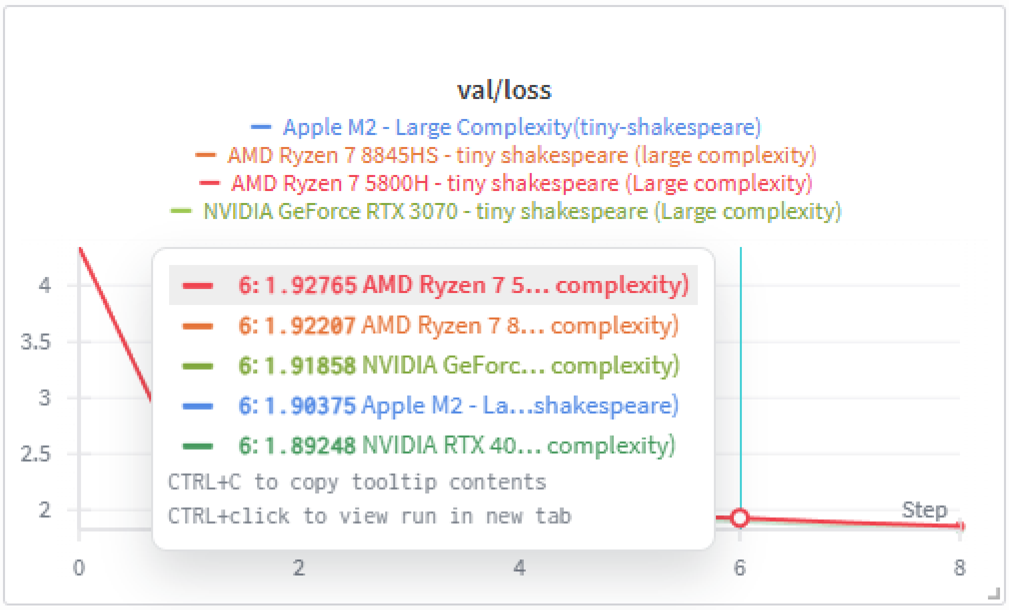
Across all hardware configurations, **training loss and validation loss curves were nearly identical**. Minor stochastic fluctuations were observed between runs (Figures 1 and 2), but these variations were not correlated with hardware type or compute capability. This outcome is expected, as all experiments used the same model architecture, hyperparameters, dataset splits, and random seeds. Since the optimization problem itself was unchanged, the learning dynamics remained consistent across devices.

To extend this experiment beyond loss-based metrics, we introduced **total runtime** (in seconds) as an additional point of comparison to capture hardware efficiency and practical limitations. Each configuration was trained for the same number of iterations, allowing runtime to directly reflect computational throughput. Runtime comparisons across the two GPUs and three CPU-only systems are shown in Figure 3.



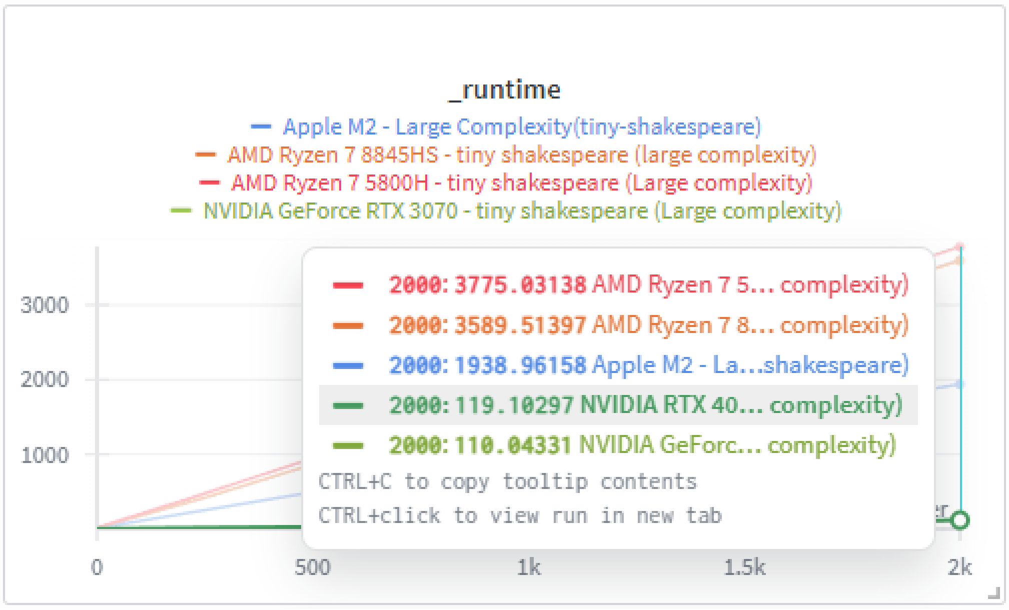
```
8: 1.70498 AMD Ryzen 7 5... complexity)
8: 1.69796 AMD Ryzen 7 8... complexity)
8: 1.69391 NVIDIA GeForc... complexity)
8: 1.69274 Apple M2 - La...shakespeare)
8: 1.68704 NVIDIA RTX 40... complexity)
```

▲ Figure 1



6: 1.92765 AMD Ryzen 7 5... complexity)
6: 1.92207 AMD Ryzen 7 8... complexity)
6: 1.91858 NVIDIA GeForc... complexity)
6: 1.90375 Apple M2 - La...shakespeare)
6: 1.89248 NVIDIA RTX 40... complexity)
CTRL+C to copy tooltip contents
CTRL+click to view run in new tab

▲ Figure 2



2000: 3775.03138 AMD Ryzen 7 5... complexity)
2000: 3589.51397 AMD Ryzen 7 8... complexity)
2000: 1938.96158 Apple M2 - La...shakespeare)
2000: 119.10297 NVIDIA RTX 40... complexity)
2000: 110.04331 NVIDIA GeForc... complexity)
CTRL+C to copy tooltip contents
CTRL+click to view run in new tab

▲ Figure 3

Notably:

- The RTX 4060 occasionally achieved slightly lower loss than the RTX 3070.
- Both AMD Ryzen 7 CPUs performed comparably to both GPUs in terms of train/loss and val/loss but did not do well in total runtime.
- The M2 Chip performed astoundingly better in runtime than both AMD Ryzen 7 CPUs
- Differences in loss values were minimal and well within expected stochastic noise.
- Compared to the second experiment, we noticed that the train/loss and val/loss both went up by around 0.09 and 0.08 respectively.

At this scale, the model converges smoothly and predictably, and hardware advantages in parallelism did translate in this case into faster outcomes but not into improved value optimization outcomes since both GPUs and CPUs performed similarly in terms of train/loss and val/loss.

These results indicate that for small transformer models with large depth and moderate learning rates, the optimization landscape is stable enough that different hardware platforms converge to essentially the same solution.

Interpretation

The results reinforce a critical distinction between **training dynamics** and **training efficiency**.

Because the **model architecture and optimization problem were identical**, the loss curves remained unchanged across hardware platforms. This confirms that **hardware does not affect the direction or outcome of optimization**, only the *speed* at which that optimization is executed.

Introducing runtime as a metric reveals the **practical limitations of CPU-based training** as model complexity increases. While CPUs are capable of executing the computation correctly, they lack the massive parallelism required to efficiently handle transformer workloads with large embedding sizes and multiple attention heads. GPUs, by contrast, exploit parallel matrix operations, allowing them to scale more gracefully as model size increases.

At this scale:

- Gradients remain well-conditioned
- Floating-point precision errors are minimal
- Attention operations are stable
- Optimization follows a deterministic trajectory

As a result, **hardware differences do not manifest in loss divergence**, but they **strongly affect feasibility and iteration throughput**. The experiment therefore highlights a clear **transition zone**: a model size where CPUs remain viable in theory but become impractical in practice due to excessive runtime.

This demonstrates that while high-end hardware is unnecessary for correctness or convergence in small-to-medium character-level models, it becomes increasingly important for **experimental velocity**, **iteration speed**, and **scalability** as model capacity grows.

Next Steps:

While training and validation loss remained identical across hardware platforms, future work will shift toward **throughput-based evaluation** to better quantify hardware efficiency. In particular, we plan to measure **tokens processed per second** as a function of training progress to expose performance divergence not visible through loss alone. To do so, we will:

- Log tokens-per-second during training using consistent batch and sequence lengths across all devices
- Plot throughput as a line graph over training iterations or wall-clock time in Weights & Biases
- Group runs by hardware type (CPU vs GPU) to visualize sustained computational efficiency
- Compare throughput stability and scaling behavior as model size increases

These additions will allow us to distinguish between identical optimization outcomes and fundamentally different execution efficiency across hardware platforms.

Conclusion

This experiment shows that when training a large-depth, large-capacity transformer, differences in hardware specifications (GPU tier vs CPU-only systems) do **not meaningfully impact model convergence or final loss values**. However, parallelism provided by GPUs does have a dramatic effect in output time. At this scale, the optimization problem is sufficiently stable that all systems reach similar solutions, regardless of computational power.

These findings reinforce the idea that **hardware advantages become relevant only when numerical instability, scale, or optimization difficulty are introduced**. For introductory models and small-scale experiments, accessible hardware is sufficient for achieving equivalent learning outcomes but they may take longer to produce these results.